Active Defense through Deceptive Configuration Techniques

GIAC (GSEC) Gold Certification

Author: Nathaniel Quist, qcuequeue@gmail.com
Advisor: Hamed Khiabani, Ph.D.
Accepted: January 21, 2016

Abstract

Security architecture, based on deception strategies can effectively change the starting point of the legitimate network, providing additional time to prepare adequate defenses. Through configurable logical networking, a virtual network can be designed to monitor, capture, and lure malicious activity deeper into a virtual labyrinth, away from the real network. By incorporating the use of dynamic threat lists, offered by specific security tools, signatures can be created to strengthen the defenses of the internal network before the malicious actions reach the boundary. Using honeynet architecture, the virtual labyrinth can be dynamically and continually created providing protection from attackers. Through real-time knowledge gathering of the attacker's exploitation techniques, the Labyrinth provides defenders with time to prepare effective countermeasures. In this paper, the demonstration on how the use of honeynet architecture can allow defense teams to strengthen their perimeter, by using customized dynamic threat lists created from a completely configured and monitored environment.

1. Introduction

Honeypots are making a profound impact in the security world. Their ability to infer information about an attacker's Tactics, Techniques, and Procedures (TTPs), allow defenders to configure their defenses to respond to emerging threats, capture 0-Day exploits, and identify malicious users within a network. Using the capabilities of individual Honeypots, building a perimeter using a networked configuration of these devices can provide a security team with an early warning detection system capable of providing actionable data used to defend the network as a whole. The networked honeypots paired with active monitoring and correlation techniques can allow certain SIEM tools to create dynamic threat lists and update these threat lists within the network's security defense tools providing near real-time defensive capabilities. By taking the manual implementation of data gathered, analysis and configuration away from the security team and replacing it with automation, the defensive posture of the network is elevated to operating at a significantly faster pace, allowing defenders additional time to focus on proactive countermeasures and less on reactionary, panicked or time constrained measures.

2. Learn the Environment

Knowing is half the battle (Hasbro, 1985). Today, magic bullets in the form of advanced mitigation techniques and one-box solutions are sold to any company wanting a quick and easy fix. Sadly, trial and error are often the means to success, and luck seemingly plays more of a role in keeping the network secure. An organization not actively attacked lends to the perception that the security team is doing an excellent job. However, when the inevitable happens and your company becomes one of the primary talking points on TV's most vulnerable, this seemingly "excellent security team" takes on an entirely new appearance.

Adages have been passed down through every profession, from master to pupil, and from SME to novice. With the mastery of these concepts, can quickly allow the novice to become the journeyman, and the journeyman to become the master. The simple

Author Name, email@address

adages are 'Know your craft' and 'Use the right tool for the right job.' The basic principles go hand in hand, only with firm knowledge of your tasks are you able to choose the right tool for the job, and only with the right tool can you hone your knowledge to advance the craft. For network security professionals, the craft is network defense and threat detection techniques, and your tools are IDS/IPS Sensors, Firewalls, Anti-Virus Platforms, Encryption, Computer Logs, Proxy Devices, File Integrity Monitoring, Configuration Settings, Policies and Procedures, Segmented Networks and Virtualization to name a few. These tools provide useful data sets, yet they only catch a limited portion of the activity that takes place within a network. Turning on the latest one-box-solution, or listening in on current threat intelligence feeds, does not mean security. Before security can take place, we need to know which tools we are using.

2.1. Critical assets

Several compliance-oriented structures mandate the detailed knowledge of a network. North American Electric Reliability Corporation's (NERC) Critical Infrastructure Protection (CIP) Standard 002-1 requires a listing "of all Critical Assets (i.e., facilities, systems, and equipment), even if such list is null." (NERC, 2009) This list is designed to be more than a compliance checkbox. An organization's security and network teams are expected to understand the network they are working within. "The list consists of assets that if destroyed, degraded, compromised (e.g., misused) or otherwise rendered unavailable would unacceptably affect the reliability or operability of the BPS (Bulk Power System) [read organization] as a whole." (NERC, 2009) Powerful language for a compliance requirement, but given that NERC CIP is designed to protect the Nation's Critical Infrastructure, e.g. Power stations, Water Suppliers, Finance, Manufacturing, Healthcare, Government, Transportation and Communications, the given word choice creates a more dynamic effect.

Not every network is considered part of the Nation's Critical Infrastructure, nor is every system identified as an organization's critical system. However, this should not limit every institution from following in NERC CIP's footprints. The definition of Critical Infrastructure highlights the basic requirements, "The facilities, services, and installations needed for the functioning of a community or society, such as transportation

and communications systems, water and power lines, and public institutions including schools, post offices, and prisons." (Dictionary, 2015) Substituting the word organization for community or society, allows the defender to quantify their network.

2.2. Trade secrets

While physical systems do play a critical part in network security, there is an entity within nearly every organization that trumps the physical systems. This entity is the information that resides within the physical systems on the network. Systems hold information, and they are vulnerable to any number of threats, so they naturally require a significant amount of time to secure. Knowing what type or types of information a particular system holds can be difficult to ascertain, but knowing what systems carry specific types information, and knowing the location of these systems, can assist defenders to identify additional threats.

A Trade Secret makes an organization a viable candidate in the field it inhabits. It separates it from its competitors and pushes it towards innovation. The Uniform Trade Secrets Act ("UTSA") defines a trade secret as: "information, including a formula, pattern, compilation, program, device, method, technique, or process, that derives independent economic value, actual or potential." (Cornell University Law School, 2015) Information is what a company is. This data is the single most important aspect of an organization. Securing the system is critical; securing the data is vital.

3. Isolate the Real World

3.1. Traditional Defense in Depth

We have covered what types of systems make the company's critical infrastructure. We have an idea what to look for, but now we need to develop methods to isolate these findings. The biggest question we need to answer is, what is our organization and how do we secure it? Defense in Depth has been kicked around for a while, and each implementation practice contains several pros and cons. The basic principles are sound and give security professionals a platform from which to jump. The four key areas of Defense in Depth are Uniform Protection, Protected Enclaves, Information Centric, and Vector Oriented (The SANS Institute, 2015).

Author Name, email@address

These four principles, as with all compliance-oriented programs, are not designed to be an ultimate security solution. Even if followed exactly, it will not guarantee a secure network. They are guidelines, overall structures for what a security professional should focus on. The creation and the following of these guidelines provide the frameworks of security. It is the responsibility of the security team to develop and refine the given structure to provide day in and day out security.

"An important principle of the Defense in Depth strategy is that achieving Information Assurance requires a balanced focus on three primary elements: People, Technology, and Operations." (Information Assurance Solutions Group, 2000) Focusing solely on the technology without taking into account the data in those systems, the people that use them, or how it is intended to function within the environment, tends to create a Whack-a-Mole phenomenon. "The attackers provoke the maintenance of a layered defensive stance that is massive, difficult to manage, requires extensive skill sets and is extremely costly." (Small, 2011) As vulnerabilities are identified, the security team begins jumping from system to system locking down openings, and applying patches. The manpower and knowledge it takes to perform these operations within a large organization are complex to manage, and the attackers know how to exploit it. "In essence, the attackers are forcing an unsustainable posture, exhausting resources and adapting advanced persistent and advanced evasive techniques to slip right past People, Process, and Technology." (Small, 2011)

However, this in no way should diminish what programs like Defense-in-Depth, and other compliance frameworks are trying to accomplish. By far and large they all hold a basic foundation, ‘Know your environment' and ‘Develop a framework to identify and respond to incidents'. Defense-in-Depth is not a magic bullet. The layered approach to security works well for a world built upon the laws of nature and with everyone treating a network as designed. But how do we define what is, and what is not, possible within a network? This question prompts the intended design to be thrown out the window. What we have to protect us is not enough to stop the most sophisticated of attackers. Our only defense is to re-learn how the network functions on a continual basis while at the same time keeping an eye on how we want the environment to behave. We need to find where the system is weak and develop how we will reinforce that weakness.

Author Name, email@address

3.2. Where legitimate traffic exists

"The Cyber-World is rife with anomalies, bugs, gaps and holes that allow an attacker to disguise traffic or even make the traffic invisible; simply passing straight through People, Process and Technology." (Small, 2011) The best way to prevent much of this risk is to dictate what is, or should be, normal. There are several options available to security teams for specifying, or at least, understanding legitimate traffic. Network diagrams, Process and File Control Lists, and Network Baseline behaviors are but a few of the tools available. Documenting the network architecture of the environment, via network diagrams and pairing that data with a process control list for each system, allows security teams to know where physical and logical systems located as well as what processes those systems are designed to use.

Network diagrams are not new. They illustrate the connectivity of each device, and they reference the physical and logical location. The physical location of the system should be printed clearly in the diagram giving information about the system, i.e. which rack houses the system, the placement within the rack, and from which power supply it draws power. The logical diagram represents which network segment a system is assigned, the VLAN it is a part of, and which systems reside in the network. This information is the building block upon which all other security fundamentals should rest. They supply information vital to production operation, disaster recovery, utility cost, hardware and connectivity constraints, and available resource capacities. The diagram must be continually updated to reflect any changes in device location. Having accurate location information on systems can alleviate time constraints when responding to incidents.

Process and File Control Lists go hand in hand with baseline system operations but differentiate a focus on what a system should be allowed to do. Process Control Lists indicate what services are required to perform a given function. A Windows 2012 R2 Server requires a specific set of services to perform its basic functionality, by adding Roles and Features to the server, it mandates additional service requirements to run on that server. By first understanding the given function of the server, the security team will be able to create lists of what each system is supposed to use during production. If that

Author Name, email@address

system also requires the use of specific file directories or network file share devices, a File Control List can be created to isolate those files or directories. By using process and file control lists, the security team will know the design, purpose, and scope of each system on the network. In turn, they can use these lists as an operational System Level Baseline.

Maintaining a network traffic baseline is critical to understanding the network behavior of a particular system or set of systems. The type of protocols they use, the average daily or hourly amount of data transferred, and how systems typically communicate with other systems. The purpose of a network baseline is to create a quantitative value for aspects of the environment that seem to alternate depending upon time or function. Network connections between systems are not a static value. However, the types of communications between systems can remain regular. A web server typically uses the protocols HTTP (80) and HTTPS (443) as part of its normal behavior, where a file server may use SMB (139 or 445) or CIFS (445) for its communications. (Microsoft, 2006) By isolating which systems require specific communication channels, this behavior becomes quantifiable. There are some systems, as well as some users, who generate more information than others. Attempting to isolate a quantitative value for how much traffic a particular system, user, or network, transmitted is challenging to ascertain, as there are variables that alter the results. However, this does not mean that you could not use a Network Baseline to develop thresholds for excessive or under-performing levels of network traffic.

Keeping the focus on the quantitative values of a network allow the security team to maintain a relatively firm hold on what is real. Creating an environment where each system has a known function or defined operation within the framework, keeps the security team on the offensive, by monitoring instead of reacting. It is critical to know the environment and how to identify when a system is functioning outside of the threshold. These are the building blocks of defense success.

3.2.1. Traditional network security stops here

Everything up until this point is considered the typical implementation within the standard Computer Security Infrastructure. The school of thought is to keep what is

Author Name, email@address

important in the network and to build a layered approach to securing the network as it stands. From this point out, a subtle change will take place. It is not the intent to abandon everything within the industry, as the principles are sound. The basic framework is to try to understand the network and secure it with Defense-in-Depth Strategies, the latest firewalls, the most current Anti-Virus knowledge bases, log correlation, and using the latest Intelligence Threat List values. The current network security measures need to stay in place, as we have not reached their true capabilities.

3.3. Configure Away from Default

3.3.1. Where are default configurations

We will now take an extra step and start changing default settings into something we dictate. This is nothing new, and we will not need to re-write the Internet to make any of this happen. The following guidelines are part of every network device or piece of legitimate production software. If it is not, take note of the limitations and attempt to convince management that a different version or vendor will allow you to alter default network connectivity values, and allow you to start taking control of the environment. I need to express that these changes may be highly controversial for some organizations as they can impose overhead on an organization's network and security teams in the form of troubleshooting and the potential degradation of its security posture.

As a tried and true caveat it must be stated, no security tool, procedure, strategy, or policy will ever be the magic bullet to end all malicious activity in a network. By changing the default network values, you can increase the complexity required to move through an environment that is expected to be default. Granted this is defined as Security through Obscurity, which is by no means a solid foundation for network security on its own, but by changing these values, we can dramatically improve the chances of not falling victim to simple attacks or at least prevent mild attacks from spreading rapidly in the environment. Additionally, coupling these obscurity methods with dedicated monitoring of attempts using default configurations does provide an advantage in identifying the threat and taking steps to securing your network against the threat.

Author Name, email@address

3.3.2. Making highways

The first change is the Default Gateway Route, referred to as the Gateway of Last Resort (Cisco, 2005). This process allows the network and security teams to isolate how routers and switches will pass data within their local network to the larger network outside. This setting determines which devices the switches and routers directly communicate. The change forces these devices to dump their routing tables and default all routable traffic to a particular device, or set of devices, mandated by what is called the Access Control List (ACL). An analogy to describe this change is, building the transportation infrastructure of the network, the interstates, highways, and parkways of the network. The value to the security team is they will be able to monitor the network chokepoints and communication intersections using IDS/IPS sensors or Network Monitor appliances to uncover potentially malicious traffic.

In the same instance, defining Default Routing Protocols can have a similar effect on a network as with changing the Default Gateway. For example, instead of relying on each vendor or application provider to define which protocol paths a particular device will use to communicate with the larger network, we can configure these devices to use only the communication paths we choose (Cisco, 2005). The default nature of every organization and home network uses a default listing of protocols to accomplish specific tasks. Malicious actors and malware designs count on these default communication ports when planning attacks. Altering these ports to communicate over a custom port directly interferes with these attack efforts. If you envision a network that is no longer using port 445 to establish their SMB share connections, a botnet configured to identify and exploit shared file systems via port 445 will not be able to spread within that network. Granted, this is not a foolproof security plan as the botnet would simply need to change the default port with the customized port configured to establish SMB Connections and it can continue its migration. However, if the security team is monitoring for default SMB Connections over port 445, they can be tipped off that a system is attempting a potential SMB migration. The defenders can start the quarantine process before the attacker can research the issue and alter the port. This type of configuration strategy allows the security team to keep events localized and control the infection before it spreads.

Author Name, email@address

There are several types of communication ports consistently kept at default settings within a large number of organizations. Remote Desktop Protocol (RDP) Connections are a common intrusion vector for many attacks. RDP connections by default use port 3389 and are commonly targeted by amateur attackers and botnets. By changing this default port to another unused port in the environment, you can prevent a large number of simple attack vectors. Virtual Private Network (VPN) connections are another highly targeted attack vector assisted by changing their default communication ports. Depending upon the type of VPN connection used in the environment several alternate protocol paths that can be used to obfuscate the VPN connections. Internet Protocol Security (IPSec) uses ports 50, 51, UDP 500 and 4500 depending on configuration, Secure Socket Layer (SSL) uses TCP 443 for web traffic, Layer Two Tunneling Protocol (L2TP) uses TCP 1701, and Point-to-Point Tunneling Protocol (PPTP) uses TCP 1723. Each of these tunneling protocols can be altered to use a different communication port for their standard operation (Juniper Networks, 2012).

Additional network tools used extensively are SMTP/Mail connections, typically using ports 25, 465, 993 and 995. HTTP and HTTPS connections transmit over ports 80 and 443, where Secure Shell (SSH) uses port 22. Virtual Network Computing (VNC) uses TCP 5900 and is starting to use TCP 5800, and UNIX VNC routinely communicates over TCP 5901 and 5801. Database connections such as SQL Servers use 1433 and 1434 for admins, and Oracle uses a multitude of ports depending upon installed components, but the most common port are TCP 1521 and 1630.

Each tool will have a unique method for altering their default values, and any tool worth installing in an enterprise environment should allow you to change their network connectivity settings. Microsoft provides a tool free of charge called "Fix It". Fix It can be used to change any number of Microsoft service network connection settings. Typically Microsoft's default configurations changes take place within the Registry settings of the host system. Fix It uses a GUI interface to assist in making these changes. Firewall providers Cisco, Juniper, and Palo Alto also have detailed documentation and tools designed to alter the default VPN connections for their appliances.

Author Name, email@address

There are additional system changes that can be made to assist in monitoring default settings. Malicious users entering an environment do not want to attract attention nor do they wish to leave traces of their tampering with the system. Cleaning system logs is a good way to ensure they cover their tracks. By moving these system logs to another system or by changing their default storage location, we can complicate their ability to cover their tracks. To add a layer of deception, keeping the default log in place but having it only be a copy of the system log can allow the actor to actually "clean" the log but they have only cleaned the fake file. A very effective means of maintaining operational oversight and allowing the attacker to tie their own noose.

Actors are also continually on the hunt for administrator credentials and alternate user passwords to allow for additional access to the local system or other systems in the network. It goes without saying, but we must change administrator passwords! However, there are additional deception techniques that can be implemented to disguise an administrator account. The original local administrator account should be modified to have little to no access while a separate administrator account can be created to fulfill the administrative functionality. The default Administrator account remains on the system, and any connectivity attempts to this account should be monitored and alerted. Providing another situation for the security team to be tipped off in advance before an administrator account compromise gets out of hand.

Malware represents another vector in which we can alter default values to provide additional monitoring tip off locations. Malware commonly attempts to bury itself within the OS directories hiding it from detection. Server teams typically spin up new systems via a template for what should be on that particular system, and using the pre-created system baseline could greatly assist in identifying changes. By partitioning the C drive to what the OS needs to function and by providing a separate partition or external file server specifically for the system or for the system's users, this creates an environment with limited and restricted read and write capabilities for the user. If the C drive is off limits to the user, a botnet or rootkit running under that user's account will have nowhere to turn. Any attempts by the user to write to the C drive are identifiable. Granted not all access to the C drive can be revoked to the user, in these situations using File Integrity Monitoring

Author Name, email@address

through a dedicated FIM provider can provide a means to detect any attempts to Read, Write, or Execute a file considered off limits.

Bridging a topic into this section that was mentioned briefly before. These changes are simply Security through Obscurity. Meaning they do not secure any device, application, or tool used within an organization, they merely change their appearance. Any persistent threat would be able to listen to the open ports on a particular system and be able to adjust their attacks to meet these new communication connections. However, this tactic does provide security through observation and uses those observations to configure your security devices. Even though the system is not more secure, we have established the clear running lanes of what we expect to see within our environment. Anything running outside of those lanes can be spotted and action taken. If an outsider enters an environment and does not know the processes and procedures of that environment, they stand a much higher chance of being caught. All we have to do is monitor for these telltale signs and then use the standard incident investigation methods currently in place to correct or remove the cause of the incident.

4. Creating the Labyrinth

The goal of this paper is to provide a method to assist the Security Team in protecting the organization from what is typically called the Advanced Persistent Threat (APT), a term frequently tossed around in Security Conferences and after any large-scale breach of a network on the public news. The APT often originates from an unknown source, and the actors are considered highly sophisticated in their Tactics, Techniques, and Procedures (TTPs). APTs are assumed to be part of nation-state organizations that hold the necessary resources to support the research, troubleshooting, and educational opportunities to make attacks successful. Due to the higher demands to successfully penetrate and avoid detection, APTs are typically stealthy, highly educated, and above all patient.

Given this threat, this paper is dedicated to organizations to help create and develop a security framework to identify even the smallest defect or change in the environment. This security framework is called the Labyrinth. The definition of a

labyrinth is a "Place constructed of or full of intricate passageways and blind alleys; a maze." (Merriam-Webster, 2015) The Labyrinth proposed is created using a networked integration of honeypots and firewall boundaries. Placed logically between an organization's ISP Gateway and the internal network of the organization. It can be placed logically in front or behind the organization's Demilitarized Zone (DMZ), where all ingress and egress traffic is forced to transit.

The Labyrinth is constructed to use the strategies of deception and information gathering. It is designed to resemble a legitimate network in almost every detail. It is logically constructed to have every system and every network path be heavily monitored and highly regulated. But perhaps the biggest functionality it will have is the use of a kill switch. The Labyrinth must be capable of cleaning and reconfiguration at a moment's notice. The Labyrinth acts as a filter for all network traffic and must have several levels of layered filtering. Possessing redundant forms of sensors to ensure the identification of all network traffic.

Through the use of custom network routing tables and proxy devices every piece of traffic must be filtered, scanned, and subject to testing. In its design, the Labyrinth will use several virtual and physical systems to construct a highly regulated maze in which any number of potential infection zones and tempting network targets. If legitimate threats do enter the Labyrinth, their actions to gain footholds within the network will be recorded, and that information can be used to protect the real environment from these same threats.

Honeypots and honeynets are present within several organizations, used as systems that lure attackers in an attempt to capture tactics and techniques. Considering there are no legitimate reasons to enter honeypots, any actions taken can instantly be held suspect. Where the Labyrinth differentiates itself from standard honeypot networks is that legitimate traffic is mandated to traverse the Labyrinth. The legitimate traffic will follow a defined path, and the unwanted traffic will face obstacles, several viable routes, no keys with which to traverse the Labyrinth, and no clear direction on where to go. As the illegitimate traffic probes deeper into the Labyrinth, it will be heavily monitored and

Author Name, email@address

recorded using industry standard network sniffers, firewalls, and systems logs. These, in turn, are fed to the security's teams SIEM solution.

A SIEM solution that offers the capability to perform automated scripted actions in response to a triggered alarm is the cornerstone for a successful Labyrinth. An automated scripted capability is either a manual, or automatic, push of a scripted function in response to a triggered alarm within the environment. For example, if an alarm were triggered that indicates a system scan from an outside source, a scripted action, paired with a SIEM alarm, can write the source IP Address of the origin system to a threat list. This threat list can then be used to identify additional systems targeted by this identified source, both now and into the future. Since the action is a scripted action, it can be created to perform any number of actions the defender scripts. The defender can create an action to place the offending IP Address on a blacklist within the organization's firewall, stopping all future connections from this IP Address.

4.1. Making Fool's Gold

We need to ensure we maintain both the attacker's interest as well as their acceptance that they are attacking real targets. Honeypots have often been criticized for their lack of believability, causing many attackers to recognize the system as fake and avoid interaction. If we allow this to happen, the Labyrinth could provide no additional intelligence on the attackers, or on their tactics and techniques, and wouldn't allow for any additional timesaving afforded to the security team.

So to begin, what is a Honeypot? A Honeypot is a non-production system, typically housed within a virtual environment, whose sole purpose is to be a target (TopSpin Security, 2015). The Honeypot is a decoy system that provides a deceptive layer, shifting the focus of attackers away from production systems. The objective of a Honeypot is to provide security teams with information about its every function, so the team can determine the tactics and techniques of those actors who interact with the system.

To convince the attacker that the decoy system is genuine, "we need to be aware that he will be trying to fingerprint the decoy and its applications." (TopSpin Security, 2015) It is critical that honeypots do not stick out and cause an attacker to move in a

Author Name, email@address

different direction. Within an entire network of honeypots, even if an attacker does move in a different direction it will likely be towards another honeypot. But, the goal of the Labyrinth is to be as believable as possible, to ensure we keep the attackers focused on what we want them to focus on, and not allowing them to question their actions.

To assist with the believability of the Labyrinth, it should behave like a real network. Each system within the internal network should have representation within the Labyrinth, and the topologic layout of the Labyrinth should essentially mimic that of the real network. To ensure the acceptance of the Labyrinth, a logical combination of honeypot systems within each subnet should resemble what the internal network would use. For example, within the DMZ section of the Labyrinth, the legitimate types of DMZ systems should be present: a fake Web Server, an external Domain Name Server (DNS), a Mail Server, and perhaps a File Server (FTP) or even a Voice over IP (VoIP) server. However, if there were also unpatched Windows XP systems with large quantities of internal information, it may cause an advanced attacker to question their situation.

It is not to mean you cannot have an unpatched Windows XP system in the Labyrinth. You should use a variety of OS's as long as they represent the types of systems with the real environment. Placing these systems within subnets that resemble a functional network. Having a network within the Labyrinth that resembles an actual subnet of Endpoint specific systems, e.g. user laptops, desktops, and printers, simply make the labyrinth more believable and facilitate the attacker to wander deeper into the Labyrinth. In turn, allowing the security team to develop a much more comprehensive listing of the TTPs on the attackers. The entire point is to make the attackers waste as much time as possible to allow your teams to counter their potential attacks.

With the effort of designing and configuring the Labyrinth to be as believable as possible, there is another level of configuration that can make the Labyrinth resemble the real thing, Administrative actions within the Labyrinth. "Day-to-day changes in the environment may include adding, upgrading and removing applications, networks, operating systems, endpoints, and devices." (TopSpin Security, 2015) Creating network traffic within the Labyrinth and by essentially treating the Labyrinth like a real environment provides yet another layer of deception. To achieve this deception, we have

Author Name, email@address

several scripted actions within the Labyrinth simulating user actions. Actions like requesting DNS lookups, Web Traffic, file transfers, and generating events that trigger log population on the systems. Creating what appears to be legitimate noise within the Labyrinth and aid in the believability of the Labyrinth's functionality. In turn, this can allow the attacker to interpret the network as normal and continue to probe and test the Labyrinth as they would any other network.

The beneficial security information gleaned from a honeypot is without a doubt the primary motivation for their use. However, there are limits and restrictions under which a Labyrinth environment should operate. "A primary concern for honeypot designers is that of an attacker getting control over it. If this happens, the attacker can initiate attacks from the honeypot, which is regarded by the network as a secure environment." (TopSpin Security, 2015) While the primary purpose of the Labyrinth is to record and use the tactics and techniques of an attacker to better secure the legitimate network, the risk of having the attacker use the Labyrinth for malicious purposes against other sites are grounds for significant legal concern. The monitoring of suspected malicious actions within the Labyrinth is the primary goal of the Labyrinth. While ultimately protecting the legitimate network is the objective, protecting outside organizations from attacks based within the Labyrinth is equally as important.

Should all identified malicious actions be stopped immediately? Within the Honeypot community it is loosely understood, that "it is sometimes better to observe the attack through to completion and then identify the stolen goods after the deed has been done." (TopSpin Security, 2015) By keeping the attacker focused on what you want them to focus on, you gather information. You are also safeguarding the attacker from actively focusing on something, or someone, else. Since the Labyrinth can be wiped clean at a moment's notice, a critical event or an action targeting an outside entity, the malicious action can be stopped in its tracks as the Labyrinth is reset to a clean version. The defenders could fail to gain a complete picture of the attackers TTPs, but the information gathered from these actions still allow the security team to bolster their defenses.

Author Name, email@address

4.2. Deception in Depth

4.2.1. Build the Labyrinth with industry standard procedures

In the vein that the Labyrinth is not designed to replace any aspect of the traditional security framework, it is to augment the information layer within the Defense-in-Depth paradigm. The Labyrinth's primary purpose is to gather information while its secondary function is to provide time to the security team in the form of delaying and deceiving the attackers. The time afforded to the security team can be used to analyze the information gathered and then to translate that information into better protection for the real network. Obscurity is the added layer of security when incorporating detailed monitoring and analysis. It allows the security team to monitor for what should not happen and prepare defenses against it.

As mentioned before, Honeypots are not a new phenomenon amongst the defensive security posture. Any number of large or distributed IT network organizations may have hundreds, or even thousands of honeypots spread throughout their networks, including local and remote systems, executing different services, operating systems, and applications (TopSpin Security, 2015). The organization of these honeypots into a dynamic network of deception-oriented devices is a relatively new concept. The increase of resources currently available as a by-proxy event with the shifting of infrastructure to the cloud; ESXi and Xen Servers are becoming increasingly available to provide Labyrinth functionality.

The location of the Labyrinth can also be manipulated to scatter Labyrinth aspects throughout critical boundaries or intersections within the organization's internal network. You could think of this idea as placing mini Labyrinths between network boundaries within the real environment itself, between internal endpoints and File Servers or Domain Controllers, between contiguous endpoint subnets, or within various parts of the DMZ network. If the internal mini-Labyrinth were found to have malicious actions within them, this could indicate the actors made it through the first Labyrinth and the internal network structure could be compromised in one or more areas.

Author Name, email@address

4.2.2. Knowing too much can hurt you

Getting the honeypot to look good on the outside and also contain something ‘sweet' on the inside requires more thought and planning than trying to catch flies (TopSpin Security, 2015). The implementation of a honeypot is not as simple as spinning up a virtual system and placing it in an easy to reach location. Without having a design in what it should look like, the types of data it should contain, or the functions of applications housed within it, the honeypot can be an obvious lure for almost any attacker. On the flip side of this coin, if the defenders are too cautious with the honeypot and its corresponding data, the honeypot may not hold enough lures to draw the attackers to it. The creation of a honeypot and its placement within the network can be called more of an art than a science. Defenders who know too much about the purpose of the honeypot may withhold a lure while a defender that knows too little may make it too appealing.

4.3. Clearly Define the Legitimate Path

With every puzzle, there must be a solution. The Labyrinth is a maze. It is a series of networking and system administration twists and turns that are designed to keep those who enter scrambling for the correct path. It is also a maze in which all legitimate network traffic to and from the internal network must transit. Defining, yet keeping the path hidden becomes essential in managing the Labyrinth. Imagine the Labyrinth as you would any properly stratified computer network. It holds the same basic components. It has a series of Firewalls that block and allow traffic to different aspects of the Labyrinth. These firewalls act as gatekeepers into different networks within the Labyrinth. Behind, or surrounding each Labyrinth firewall are network traffic sensors monitoring for traffic, keeping a watchful eye on all communications attempting or making it through the firewall. The firewall and sensor pairs that do not align with the legitimate path should receive little to no traffic traversing their barriers, allowing for a higher granular network packet inspection.

The firewalls that align with the correct path should have network traffic sensors placed logically placed with the flow of traffic. They should be configured to have a layered series of network traffic inspections routines to investigate for a particular type of

network traffic or attacks. For example, the first boundary inspects for flood attacks, or fragmented packets or syn attacks. The consecutive boundaries will monitor for email traffic, XSS attacks, deep packet inspection, and VPN attack vectors. By placing the various Internet traffic analytic tools in line with the only available egress and ingress points, this can assist in the security team's ability to identify a suspicious event. Examples of these analytic tools are IDS/IPS, email inspection, Layer 7 Firewalls, Proxy or Network Address Translation (NAT) devices, and antivirus detection appliances.

The key to allowing legitimate traffic through the Labyrinth is the sequence in which traffic is forced to travel. This variable is dynamic and can shift as all components of the Labyrinth are subject to the network configuration prescribed by the Labyrinth template. Legitimate traffic could be forced to navigate through four different firewalls, where the Labyrinth itself could house double or triple that amount to make the Labyrinth as maze-like as possible. These four firewalls could then be altered at random to keep the sequence of firewall gateways obscured.

The correct path through the Labyrinth will be set initially within the Labyrinth template. The templates house all routable information needed by the Labyrinth firewalls to pass legitimate traffic. They also house the firewall configurations, as well as, the settings for network transmission for Labyrinth logging and network traffic monitoring requirements. Since no two Labyrinths can be identical, the need for a detailed template that houses all of the routable traffic and system configurations needs to be highly vetted to ensure no vulnerabilities exist that could allow an attacker to skip the Labyrinth. The more detail given to the Labyrinth templates, the more secure the Labyrinth will be and the faster the Labyrinth will be able to cycle between templates.

5. Watching the Maze

5.1. Building Sentry Towers

If the monitoring system can distinguish events generated by attackers from those originating from legitimate activity, this helps reduce the amount of effort that the administrator must expend in analyzing attacks (Asrigo, Litty, & Lie, 2006). By stratifying the network sensors along the legitimate path through the Labyrinth, the

sensors are better able to act like Sentry Posts, scrutinizing over particular aspects of the flow of data. It is a known fact within network traffic analysis that a single sensor cannot be expected to collect and analyze every packet crossing the network for every type of potential threat and still maintain an acceptable network speed.

Sensors require a stratified approach to ensure they have enough resources to perform a complete inspection of the network traffic. One sensor is looking for user threat vectors while another sensor is analyzing potential SQL Injections or covert protocol attacks. Relying on one sensor to perform analytic operations is illogical and results in poor security. IDS/IPS Sensors, Next Generation Firewalls, Log Collection and Packet Recompilation need to take place at various levels between the ISP and the legitimate gateway to ensure they are capturing all of the potential malicious traffic. Using the term Sentry Tower in place of a sensor, or a capture point turns the concept of deep packet analysis into a physical place of monitoring. They are positioned to separate the flow of traffic into a manageable trickle with each Sentry Tower inspecting for a particular piece of the overall security footprint.

Each Sentry Tower is assigned a particular collection system that it will gather all of the logs, potential sensor hits and aggregated event data that will be analyzed by the IDS/IPS Management Systems, Centralized Logging Systems, and the SIEM solution. Correlating the gathered data into actionable and recordable events allows the management system to report findings and provide assistance during analysis. The Sentry Towers are essentially the first line of defense within the Labyrinth for separating traffic into manageable sections ensuring the identification of each packet as well as diminishing the overall speed through the Labyrinth.

5.1.1. Watchtowers

By incorporating an additional level of inspection within the appliances that host the virtual systems, provides for a lower resource requirement when monitoring traffic. This investigation is available within the system that hosts the virtual systems themselves. The term Watchtower gives these systems a more broad definition, as these systems provide a higher level of system monitoring since they are monitoring the traffic and functionality of the virtual devices they are hosting. The term hypervisor is

Author Name, email@address

commonly referred to as the hosting layer of virtual appliances, even though this is only fitting to one type virtual system hosting.

These Virtual Hosting systems come in a variety of flavors, all falling within the term Virtual Machine Monitors (VMM). By placing sensors on the virtual system host, it provides an ability to monitor the virtual system functionality within a strongly isolated environment. Collected information from these systems can provide insight on tampering, but at the same time, give the individual sensor full visibility into the system (Asrigo, Litty, & Lie, 2006). If we "analyze and categorize the attacks detected by our system, and find that by monitoring for actions that attackers take after a compromise, rather than monitoring for exploitation of a vulnerability, we are able to detect a large number of attacks with relatively few sensors." (Asrigo, Litty, & Lie, 2006) This ability to monitor a larger segment of the Labyrinth with fewer sensors translates into a lower resource cost for organizations to implement a Labyrinth.

To achieve a lower resource cost, the monitoring the Kernel level processes of the hosted virtual machines proved to be the most beneficial. Any alteration of the virtual system's kernel is a direct symptom of tampering or compromise of the system. As there are no SIEM or anti-virus agents installed directly within the virtual system's directories, it creates a juicer target for attackers that may be infiltrating the Labyrinth.

The VMM detects changes within the kernel of the Virtual Operating System through the use of Virtual Machine Identifier (vm_id) tags passed to the VMM from the Virtual System. The VMM passes the vm_id of the virtual machine that triggered the event and then sent to the SIEM for correlation (Asrigo, Litty, & Lie, 2006). With the ability for the VMM to monitor, record, and transmit these vm_ids from all of the virtual machines it is hosting, this allows each virtual machine to process its local data, without needing to host monitoring tools. Allowing a single VMM sensor to monitor multiple virtual machines simultaneously (Asrigo, Litty, & Lie, 2006).

With the focus on VMMs, there are two variants that I will focus on, the Unified Modeling Language (UML) and the Citrix Xen Server Virtual Machine Host Appliances. The UML VMM is likely the most common type of individual virtual machine hosting that most computer security professionals are familiar. UML is the backbone for

Author Name, email@address

VMWare, VirtualBox, and Parallel Virtual Machine hosting. UML requires a host Operating System to run an application, which houses another Operating System. The best term that encapsulates this process is that of Emulation. "UML uses three host operating system processes to emulate a full virtual machine. One process runs the guest kernel while another one runs the guest process in the virtual environment. Finally, the third process emulates I/O and block device operations." (Asrigo, Litty, & Lie, 2006) These three processes function together to allow multiple Operating Systems to run simultaneously within one sufficiently powered personal computer or server.

Xen, on the other hand, is an unhosted VMM. Meaning, the presence of a Host Operating System like Windows, UNIX or Linux is not available. The Xen configuration allows direct access to the hardware from the virtual systems. The term Hypervisor relates to this type of VMM where Xen essentially brokers the connection requests from the virtual machine and the hardware. The monitoring of the Virtual Machine's Kernel takes place with the requests on hardware, and as is similar to UML, the vm_id of the events can be recorded and sent to the SIEM for correlation. Xen has a beneficial impact on resources at this level of operation as it allows for a lower resource hit on the hardware itself, as a host Operating System is not required. This method of Virtual Hosting is in production within the following types of virtual appliances, VMWare's ESXi, Citrix's XenServer, and the Kernel-based Virtual Machine (KVM).

5.2. Monitor Traffic

The method of monitoring and recording events within the Labyrinth follows the same principles as recording events within a network. Operating Systems, Network Devices, and Firewalls send their logs to a centralized location, like a SIEM. Within this centralized logging platform, the SIEM can correlate the traffic using various metadata fields to ascertain an observed or statistical pattern. Each device within the Labyrinth will send all of the logs or captured network traffic to the SIEM. To more accurately and quickly identify events, these event details can then be used to perform secondary functionality.

If the honeypots are hosting Windows Operating Systems, then the standard set of Windows Event Logs can be gathered from the virtual system via a locally housed SIEM

Author Name, email@address

Agent or they can be pulled remotely from the Virtual System via a similar style of SIEM collection agent, which will ferry the logs to the centralized Platform Manager. Depending upon the type or functionality of the virtual machine, additional logs can be gathered to monitor system functionality. For example, Domain Controllers, IIS Servers, Email Servers, etc. Linux and UNIX Operating Systems also store logs that can either be locally collected via the /var/log/messages directory, or they can be gathered remotely with the virtual machine sending the system logs via syslog. Being virtual machines, the option that collects these logs can also be discussed with Virtual Machine engineers to see if the VMM Monitoring is an option that best serves the needs of the security team. Typically, one or the other of these two options is the best route. Having both local log collection and VMM monitoring would be overly resource intensive and may not provide additional valuable information.

Virtual machines are not the only sources of logs that should be gathered by the SIEM. Firewalls, preferably Next Generation Firewalls, also known as Layer 7 Firewalls, provide a great deal of insight into the types of network traffic that being transmitted through the Labyrinth. Providing a deep packet inspection of network traffic allows the firewalls to uncover malicious activities attempting to pass into the network. Sending the logs of these Firewalls to the SIEM solution provides the SIEM with an additional source of information detailing network traffic behavior. The traffic can then be correlated with the interior system logs to develop a better understanding the Labyrinth's functionality.

Network traffic monitoring does not stop at the Firewall. Data from IDS/IPS sensors, Proxy servers, NAT devices and Network Monitoring devices all provide an additional layer of network traffic analysis and investigation properties. All of the data from these devices provides the SIEM with the best possible picture of the environment.

5.2.1. Listening Posts

Another type of monitoring activity can come in the form of honeytokens, also known as Word Bugs. Honeytokens have their functionality explained by John Strand and Paul Asadoorian's book, Offensive Countermeasures: The Art of Active Defense, “[inserting] HTML [embedded within a word doc], and Word will make [the html] call back to your systems (Strand & Asadoorian, 2015).” Honeytokens are documents that

have imbedded HTML code which when opened will attempt to recover the HTML information by calling back to the assigned HTML link to download the requested information. By populating the document with an HTML code to phone home, any HTML communication entering the environment would be an indication that information had left the organization's network.

The Labyrinth can make great use of this tool to better understand the attacker. When opening the honeytokens within the attacker's infrastructure, the beacon will contain its current location, which will allow the defense team to add their IP address to our block list, as well as provide the legal team within further evidence of criminal activity. By scattering several honeytokened documents throughout the Labyrinth, the security team will be able to place additional restrictions on their legitimate network to protect the network. As the HTML beacons enter the environment, the SIEM will be able to parse the source IP address from the beacon and add that IP Address to a blacklist for all ingress or egress traffic. This IP Address is then added to the block list of the internal network to ensure no communication from this IP Address is allowed through the firewall.

5.2.2. SIEM Collection

Before information can be correlated, it first needs to be collected. Every SIEM Agent has multiple methods with which it can collect data. One method is that of File Integrity Monitoring (FIM). This feature allows a FIM agent to monitor local file and directory access, modifications, and deletions in real-time. The second method of data collection comes from the local collection of system data. Operating Systems locally store Event Logs, which are gathered by a locally installed SIEM Agent and then transported to a centralized managing device. The third method used for data collection is that of remote log collection. Typically accomplished via Syslog when collecting data from *NIX variant Operating Systems, or from Security Applications, and from network devices. Windows systems can also be configured to send syslog data via a third party application.

Naturally, the network traffic itself will need to be monitored within each section of the Labyrinth as well. The monitored traffic will be collected via a span port attached

Author Name, email@address

to a switch or router. Typically the collection agent is placed behind each section's firewall. The connection of an IDS/IPS sensor to a span port relays this information to the central host. The span ports are connected to network devices within the Labyrinth immediately behind the entrance firewall, as well as behind each subsequent sections firewall, and finally placed both in front of and behind the gateway to the internal network. This traffic will then be monitored for various network traffic pattern analysis and behavior analysis as they progress through Labyrinth. All of this sensor traffic will be feed into the SIEM appliance to complete the broad correlation of the traffic analysis between all of the sensors.

6. Creating Time

Delivery of traffic to the SIEM from each device within the Labyrinth provides a well-rounded set of system and network related data. So how do we get from recording the Labyrinth data to providing actionable intelligence gathered from the Labyrinth? SANS had a poster in 2014, which read, "Know Normal, Find Evil". The job of the security teams up until this point has been doing nothing other than knowing and defining normal. In learning the structure of the organization, what critical systems are, and where critical data resides, we made changed to the default configurations. Every aspect the network should be configured and manipulated to fit the needs of the organization. All of these changes make the static functionality of the network, and define "Normal." Anything that falls outside of these configured changes can be identified as abnormal, leaning towards "Evil," laying the groundwork for the timesaving benefits of the Labyrinth.

6.1. The OODA Loop

In the 1960's, shortly after the Korean War, US Air Force Colonel John Boyd radically altered how the Air Force approached and trained their next generation of fighter pilots. Colonel Boyd was considered a maverick, unconventional, and uniquely talented when he piloted an aircraft. He was quoted saying to any newcomer at Nellis Air Force Base "He could put them on his six and outmaneuver them for a kill in less than 40 seconds." (Hammond, 2012) He was good at his job and took drastic steps to improve the

Author Name, email@address

training and capabilities of fighter pilots. He is the creator of the Military's mechanical and organic training moniker, the OODA Loop. OODA stands for Observer, Orient, Decide and Act. The very basic theory behind the OODA loop is that if you are able Observer, Orient, Decide, and Act, faster than your opponent, you will win the battle. This idea is beneficial if engaged in a dogfight 10,000 ft in the air, but the principles apply to any challenge that features opposing forces. Computer security is no exception.

Referring to John Strand and Paul Asadoorian's book, they also made a reference to the OODA loop and had this to say about it, "Many of our technologies only notify us when we are under attack. In the world of OODA, this is the equivalent of having a burlap sack over you head and having someone beat you. By the time the first blows strike, it is already too late." (Strand & Asadoorian, 2015) A very poignant phrase, and undoubtedly true. Having a tool to identify and alert the defense team when an attack has occurred, or better yet when an attack is currently taking place, is a great thing to have. But, having this information instantly puts the defenders behind the attackers in a desperate position to catch up. In the world of OODA, you failed to Observe, Orient, Decide, and Act faster than your opponent. Sadly, this scenario is only the best-case scenario. The 2015 Verizon Data Breach Investigations Report found that time an attacker has to wait as little as 22 seconds for a click on phishing scam (Verizon, 2015). The average Mean Time to Detection for most intrusions is not marked within minutes, hours or days of the infection, but within the months. Standard Tactics, Techniques, and Procedures of attackers is to download and install tools that access the system again (backdoors). They hide their presence with rootkits or attack other machines (Asrigo, Litty, & Lie, 2006). It is clear that our present methodology for performing the skills of Observation and Orientation is lacking.

The power of the Labyrinth is in its ability to create time and respond to threats. Time created by delaying and distracting the attackers and affording our defenders time to identify, create, and enable actions to defend the network. By using the Labyrinth as a statically controlled playing field, any suspicious actions on devices, within any section of the Labyrinth, can immediately be flagged as suspicious and added to threat and blacklists. These lists are then used to protect and monitor the real environment for malicious actions. The power of this security framework is that we use the speed of

Author Name, email@address

computers to assist us in our ability to secure and monitor networks. Coupled with the clearing and reconfiguration capabilities of the Labyrinth and with dynamic threat list creation capabilities provided from the SIEM, we can greatly assist a security team to provide immediate actionable data.

7. Conclusion

The Labyrinth alone will not guarantee a level of advanced security. Every aspect of the Labyrinth must be recorded and compared against a control list to isolate change. This data must then be forwarded to a centralized location to allow for correlation and to alert upon anomalies. SIEM appliances are tools that store Labyrinth data in a centralized location and allow for automated actions. Increasing the speed of securing the network via its automated scripting capabilities. Defenders must know what threats they are expecting, and they must prepare the Labyrinth and the SIEM frameworks to complete the security advantages. This process requires the OODA Loop, as the simplistic beauty of the OODA loop holds the principles of Observing, Orienting, Deciding, and Acting are never truly completed. Attacker's techniques and tactics will change. The discoveries of new vulnerabilities and 0-day threats will emerge. Innovation must never stop on the side of the defenders. By defining normal, monitoring abnormal, and quantifying what was previously considered subjective, the Labyrinth functions like a sensitive control net for filtering out anomalies. Monitoring for default actions and abnormal behavior, the SIEM can correlate, alert, and dynamically respond to suspicious events at a much faster rate than manual correlation. Defenders are then afforded the time to research new threats and to configure new threat lists that will combat the next generation of attacks.

Author Name, email@address

References

Asrigo, K., Litty, L., & Lie, D. (2006). *Using VMM-Based Sensors to Monitor Honeypots.* University of Toronto, Department of Electrical and Computer Engineering. Toronto: University of Toronto.

Cisco. (2005, August 10). *Configuring a Gateway of Last Resort Using IP Commands*. (Cisco, Producer) Retrieved December 19, 2015, from Cisco: http://www.cisco.com/c/en/us/support/docs/ip/routing-information-protocol-rip/16448-default.html

Cornell University Law School. (2015, December 18). *Trade Secrets*. Retrieved December 21, 2015, from Legal Information Institute: https://www.law.cornell.edu/wex/trade_secret

Dictionary, T. A. (2015, January 1). *Infrastructure*. (H. M. Harcourt, Producer) Retrieved December 14, 2015, from The American Heritage Dictionary: https://www.ahdictionary.com/word/search.html?q=infrastructure

Hammond, G. T. (2012). *On The Making of History: John Boyd and American Security* . US Air Force Academy , US Air Force Academy.

Hasbro (Director). (1985). *G.I. Joe PSA* [Motion Picture].

Information Assurance Solutions Group. (2000). *Defense In Depth.* Fort Meade, Maryland: National Security Agency.

Juniper Networks. (2012, December 31). *What ports are used for a Virtual Private Network (VPN).* (J. Networks, Producer) Retrieved December 16, 2015, from Juniper Networks: http://kb.juniper.net/InfoCenter/index?page=content&id=KB5671&actp=search

Author Name, email@address

Merriam-Webster. (2015, January 1). *Labyrinth*. (Merriam-Webster, Producer) Retrieved December 18, 2015, from Merriam-Webster Dictionary: http://www.merriam-webster.com/dictionary/labyrinth

Microsoft. (2006, January 1). *Securing Your Web Server*. Retrieved December 18, 2015, from Developer Network: https://msdn.microsoft.com

NERC. (2009). *Security Guideline for the Electricity Sector: Identifying Critical Assets.* North American Electric Reliability Corporation. Princeton: NERC.

Small, P. E. (2011). *Defense in Depth: An Impractical Strategy for a Cyber World.* The SANS Institute, InfoSec Reading Room. Bethesda: The SANS Institute.

Strand, J., & Asadoorian, P. (2015). *Offensive Countermeasures: The Art of Active Defense.* Lexington, Kentucky.

The SANS Institute. (2015). *Security 401.2: Defense In-Depth* (Vol. 2). (T. S. Institute, Ed.) Bethesda, Maryland: The SANS Institute.

TopSpin Security. (2015). *The Art of Decoy and Honeypotting.* DECOYnet. TopSpin Security.

Verizon. (2015). *2015 Data Breach Investigations Report.* Verizon. Verizon.

Author Name, email@address